XPRESS: A Novice Interface for the Real-Time Communication of Mathematical Expressions

Marco Pollanen[†][*], Thomas Wisniewski[‡], and Xiao Yu[†]

[†]*Dept. of Mathematics, Trent University, Peterborough, ON, Canada, K9J 7B8*
[‡]*Dept. of Computer Science, Trent University, Peterborough, ON, Canada, K9J 7B8*

Abstract

In recent years there has been a great deal of research regarding entry systems for mathematical expressions. With the rapid development of online learning, there is now a need for mathematics entry interfaces for novice users. These users often need to quickly communicate mathematical expressions to an instructor in a live online discussion, or to a server in a timed testing environment, without having had training in the use of any mathematical entry system.

In this paper we discuss the issue of real-time expression entry for novices. We also introduce a new open-source AJAX/SVG Web-based user interface for expression entry by novices, called XPRESS (Transformation of Pictorial Representation of Expression Spatial Structure). With this interface the user can quickly lay out the expression without constraint. The expression is then converted to TEX using a spatial analysis algorithm.

Key Words: Formula Input Systems, Mathematics Communication, Human-Computer Interaction, Mathematical User Interfaces, Novice User Interfaces

1 Introduction

Online learning has been rapidly developing in the last decade, aided by advances in Internet communication technologies. However, many of these communication technologies, from e-mail to instant messaging, are text-based. In fact, they can be described as *inline* text technologies, as the only spatial relationships between characters are based on the order in which they occur. Inline text-based communication introduces significant challenges to the online communication of mathematics. For example, consider the simple expression

$$\epsilon^2 \le \frac{1}{2x}.$$

A professional mathematician familiar with TEX-notation would see that this could be represented as

```
\epsilon^2 \le \frac{1}{2x}.
```

They should also have no problem in understanding the representation

[*]Corresponding Author: marcopollanen@trentu.ca

```
epsilon^2 <= 1/(2x).
```

This might, however, be much more difficult for someone without experience in entering mathematics on a computer. Even if one knows the name of the symbol ϵ, and the common conventions such as `^` for exponents, and avoids the ambiguity of writing `1/2x` instead of `1/(2x)`, one would still have to deal with the $\leq$ symbol. In the above example `<=` is used, as it follows conventions in many programming languages, and should be easy to interpret by a general audience. It could still, however, be misconstrued as a reverse implication sign. If the $\leq$ symbol had instead been $\not\leq$, one might have had to resort to writing "NOT LESS THAN OR EQUAL TO" to be understood by a general audience.

The main problem is that in addition to having a very large number of symbols, many with no commonly agreed-to name, mathematics notation is inherently two-dimensional, with spatial relationships between symbols playing a critical role. This can be seen in some simple examples a first-year university student might encounter, such as:

$$\int_0^1 \frac{x^2+2x+1}{\sqrt{x+\frac{x+1}{2x+3}}}dx \qquad \text{and} \qquad \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix}.$$

In online learning a student must often be able to communicate mathematics in real time, where there is a real or perceived time constraint. For example, a student may engage in a live mathematics conversation with their instructor or other students who might be waiting for a response. In [1] the specialized mathematics whiteboard enVision [3] was used for live online mathematics communication. enVision essentially provides an interface that allows participants to "draw" their mathematics with a keyboard and mouse. The usefulness of such a system, however, would be greatly improved if it could interact with a computer algebra system, as is done in E-Chalk [6], or if the content could made reusable by, for example, converting it to TEX.

Another situation where students must communicate in real time is when an interactive or adaptive question server, such as Xero [4], is used for time-limited online quizzes where students are required to enter free-form answers to mathematics questions. If a student has difficulty with entering mathematical expressions into a computer, he or she may be tested inadvertently on the use of the mathematics entry system rather than the extent of his or her mathematical knowledge.

Most students who take mathematics courses are not mathematics majors, and so it is unrealistic to expect them to become experts on a text-based mathematics language such as TEX. It would also be unrealistic to expect them to spend a great deal of time learning any alternate entry system. Accordingly, we will consider them to be novice users.

Most university students taking a mathematics course do not have significant difficulty with quickly writing down mathematics expressions on paper or a blackboard. Therefore, any difficulty with entering mathematical expressions into a computer can be attributed to the failure of the expression-entry mathematical user interface. Assuming that a user has a *mental image* of the expression they wish to communicate, the goal of an expression-entry system is to allow a user to express this image in a format that is meaningful to a computer, and to do so as easily and quickly as possible.

There are two main classes of visual mathematical expression editors: direct-manipulation editors, such as those found in Microsoft Word, and natural entry systems based on pen and tablet input. In the next section we will examine each of these classes of interfaces from the perspective of real-time communication by a novice user. In section 3 we will introduce a new type of expression editor, based on a hybrid of these classes, designed specifically for the novice user.

2 Visual Expression Editors

2.1 Direct-Manipulation Editors

In a structure-based direct-manipulation editor, a user directly edits a formula as it is being displayed. Palettes of additional mathematical symbols are available in menus, as well as mathematical structures which contain empty regions for subexpressions, often denoted by a $\square$ that the user can enter subexpressions into. Examples of structures are $\frac{\square}{\square}$ or $\sqrt{\square}$, representing respectively fractions and roots. These structures can be nested as needed. In such an editor, the integral from a previous example might look like this:

$$\int_{\boxed{0}}^{\boxed{1}} \frac{\boxed{\boxed{\text{x}}^{\boxed{2}}\boxed{+2\text{x}+1}}}{\boxed{\sqrt{\boxed{\text{x}+}\frac{\boxed{\text{x}+1}}{\boxed{2\text{x}+3}}}}}\boxed{\text{dx}}.$$

Users may click on any box to change the keyboard focus to it and edit the box contents or insert another nested structure inside. Alternatively, they may use their keyboard cursor keys to navigate the structures. However, navigation may, at times, be inconsistent with *what-you-see-is-what-you-get* principles. For example, let us consider the cursor's behaviour in the equation editor contained in Amaya, W3C's Web-browsing and authoring environment. If the cursor starts at the left of the above example, repeatedly hitting the right cursor-key will traverse all the substructures in the above expression, at times seeming to move in the reverse direction to that indicated by the cursor-key. This violates the user interface notion of *geometric navigation* (see [2]), according to which the cursor should move in a direction consistent with what the user would expect.

There are a significant number of direct-manipulation editors. A review of many of them and their dynamics is provided in [2]. There is a great deal of inconsistency among these editors with regard to even basic user interactions, such as navigation and editing.

There are many aspects of a structural editor that create difficulties for novice users. For example, suppose a user wishes to enter the expression $\frac{\sqrt{x}}{y}$. The default method for entering this nested structure is to first select $\frac{\square}{\square}$, followed by $\sqrt{\square}$. However, someone writing such an expression on a blackboard would likely write the expression structure in the reverse order, first completing the numerator, followed by the division line and then the denominator. Thus the user must reverse the order of entry in their head; in essence the user is required to "parse" the expression before entering it. Not only might this behaviour cause confusion, but the added step of mentally parsing the expression is dissimilar to the task of communicating the expression, and there may be some *destructive interference* between the tasks.

Another usability problem is that in many of these editors it is not easy for a user to modify expression structure. If one writes down an expression, it can be difficult to manipulate it as one would do on a blackboard. For example, suppose a student writes the intermediary step $\sqrt{\sqrt[3]{(x^2+1)^3}}$ in an online conversation. It can often be difficult to simplify the expression in a structural editor by removing just the cube root and cubic power.

In testing expression editors on users, we found that many novice users make structural decisions which lock them into a path where they are unable to complete the expression they wish to enter. This would be unacceptable if the input system were used on an online exam. A free-form entry system, which does not constrain the user to structural models, may be better suited to this application. A pen-based system is an example of a such a free-form entry system.

2.2 Pen-Based Expression Input Systems

In a pen-based input system, such as FFES [5], a user enters a handwritten formula using a pen and tablet. A character recognition algorithm is typically run on each character to identify the mathematical symbol, and then a structural analysis algorithm is used on the identity of each symbol and its bounding boxes to create an expression representation in a form such as TEX.

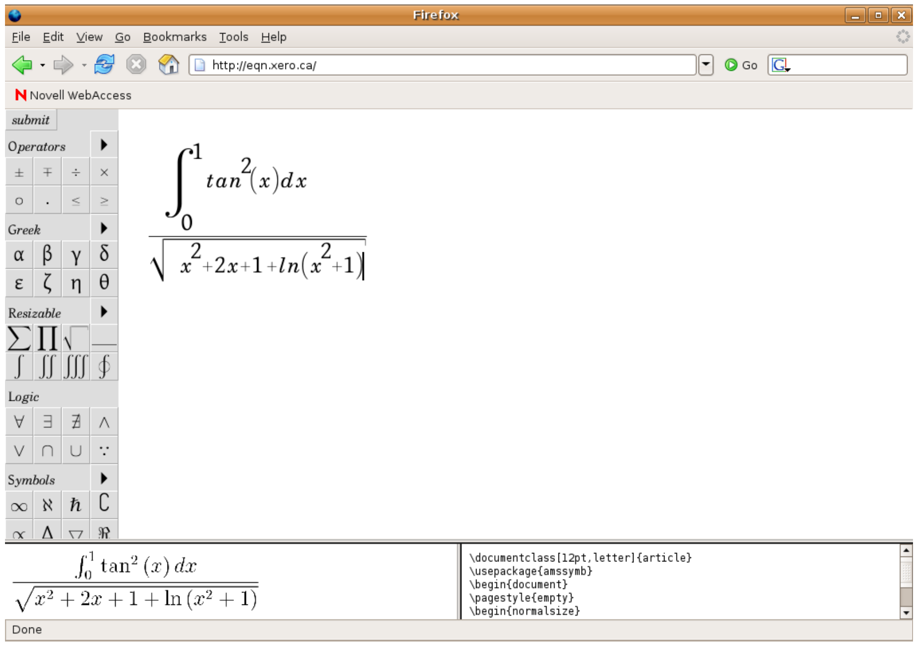
Ideally such as system would replicate the experience that a user would have with a blackboard. With current algorithms, however, failure rates for character recognition are relatively high, especially for novice users. In practice, the character recognition algorithm has to be trained on the user's handwriting, and the user must learn proper input techniques. Even then, the input interface will probably have needed to interactively prompt users to determine the distinction between many similar symbols such as $\langle$ and $<$.

A more practical consideration is that tablet hardware is not yet commonly owned by most users.

In the next section we will introduce a system that we are developing, which is a hybrid of a palette-based system and the pen-based system, and which attempts to build on the strengths of each approach.

3 A Free-Form Hybrid Input System

In this section we introduce a new hybrid input platform called XPRESS (Transformation of Pictorial Representation of Expression Spatial Structure) that is designed with novice users in mind. The underlying assumption in its creation is that a novice user has a mental image of the mathematical formula they wish to express. Thus the goal of XPRESS is to allow the user to create a computer "picture" of this formula as quickly as possibly by using their keyboard and mouse. XPRESS uses a palette-based approach of a structural editor to allow users to input symbols, and follows the free-form approach of a pen-based system to allow users to place symbols where they wish. Once the symbols are placed, a structural analysis algorithm is applied to the completed formula, which is converted to LaTeX. One advantage of this pictorial approach, as in the pen-based system, is that the "picture" created can be transmitted in a real-time discussion before the expression is complete, so that users are not kept waiting, or it can be stored with the converted expression in on online exam if manual verification is needed. In the following subsections, we will provide an overview of the XPRESS system and its keyboard and mouse interactions.



$$\frac{\int_0^1 \tan^2(x)\,dx}{\sqrt{x^2+2x+1+\ln(x^2+1)}}$$

```
\documentclass[12pt,letter]{article}
\usepackage{amssymb}
\begin{document}
\pagestyle{empty}
\begin{normalsize}
```

Figure 1. The XPRESS prototype front-end running in Firefox 1.5 for Linux. The user input canvas is the large panel in the upper right-hand corner, while the lower two panels, are respectively, LaTeX image output and LaTeX source.

3.1 Overview of System

XPRESS is an open-source package consisting of a browser-based front-end (see Figure 1 for a screenshot) as well as a graphics server. The front-end can be thought of as a diagram editor. However, instead of boxes and shapes, a user may select mathematical symbols to be placed anywhere they wish. The symbols may be moved or deleted at will, and many may be resized. The

Xpress front-end is a Scalable Vector Graphics (SVG) document supported by Asynchronous JavaScript and XML (AJAX). It is cross-platform and will run in recent versions of Firefox, Opera, Safari (development version) without a plug-in, and in Internet Explorer using Adobe's free SVG Viewer plug-in. Once the user has completed a picture of the expression they wish to represent, they press a button to submit each symbol and its bounding box information to the server. On the server this information is then analysed by a baseline structural analysis algorithm, which is modified from [7], to create a parsed expression. In our prototype, the expression is then converted to LaTeX, and both the LaTeX source code and a final image are sent back to the user. In future versions of Xpress, expressions may be converted to other machine readable formats to interface with applications such as computer algebra systems.

The goal of Xpress is to make it quick and intuitive for novice users to input and modify mathematical expressions. Thus user interactions should be built on models with which most users are familiar. As a keyboard is most often used to communicate in a "text-editor" fashion, keyboard interaction in Xpress is designed to behave as a user would intuitively expect it to behave in a word processor. Xpress mouse interaction is modelled after what a user would expect in a vector-based diagram editor, such as that found in Microsoft Word.

3.2 Keyboard Interaction

The input canvas in the Xpress front-end always displays a cursor for text input. The cursor navigates the input canvas in a geometric fashion without constraint, including empty space (i.e., the cursor moves freely in the direction of a cursor key press). The cursor acts like a cursor in a standard text editor with respect to text insertion and deletion. However, minor differences include the following:

- **Half-Steps for Exponents:** The cursor moves a half step at a time in the vertical direction to make exponents and subscripts more natural. As relative positions of symbols are used to determine their relationships, exponents do not have to be entered in a smaller font and may have a larger vertical offset than half a step.
- **Shortcut Key Mappings:** To accommodate the large number of possible symbols, whenever a letter is pressed it is inserted in the cursor location and a button panel appears near the cursor with a few related alternate symbols. The user may select an alternate symbol to replace the one just inserted by using the Tab key. For example, the related symbols for the letter a are $\alpha, \forall, \wedge, \aleph$, and $\angle$. As these symbols are related in a natural language sense to the letter a, these shortcut mappings reduce dual-task destructive interference.

3.3 Mouse Interaction

Symbols not appearing on the keyboard do not have to be entered using the keyboard shortcuts, as they are all available in panels of symbols and can be selected using the mouse. The mouse also plays a major role on the canvas in editing and moving symbols. XPRESS is designed to use only single clicks of the left button for all interactions with the canvas and to be mode-less in the sense that the user is always in edit mode. The mouse interacts in the following fashion:

- **Moving a Symbol:** An individual symbol on the canvas may be moved by pressing down the mouse button on top of the symbol, and dragging the symbol to its new location and releasing the button.
- **Selection Symbols:** A group of symbols may be highlighted by pressing down the mouse on an empty portion of the canvas and dragging out an outline. All symbols falling inside the outline will be grouped and may be dragged or deleted together.
- **Resizing:** Resizable symbols contain special resizing points which allow the symbol to be stretched in various directions.

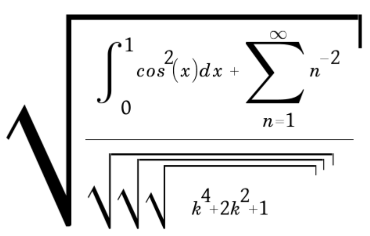
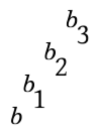
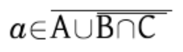
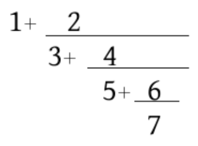
User Drawn Input	Compiled LaTeX Output
	$\sqrt{\frac{\int_0^1 \cos^2(x)dx+\sum_{n=1}^{\infty} n^{-2}}{\sqrt{\sqrt{\sqrt{k^4+2k^2+1}}}}}$
	$b^{b_1^{b_2^{b_3}}}$
	$a \in \overline{A \cup \overline{B \cap C}}$
	$1+\frac{2}{3+\frac{4}{5+\frac{6}{7}}}$

Table 1: Sample user-drawn XPRESS input and corresponding automated LaTeX output after structural analysis.

4 Conclusion

In this paper we raised issues regarding the design of mathematical expression input interfaces for novice users. We suggested that the best approach to resolving these issues is to create a novice interface that interacts in ways expected of software with which most users are familiar, such as text editors. We created a prototype interface, called XPRESS, which attempts to interact in this fashion.

Preliminary tests of the interface indicated that novice users found its use to be intuitive and were easily able to create and modify complex expressions. The usability of the interface will be studied more extensively in a first-year online calculus course in the next academic year as both an input method for online office hours and for inputting expressions in online assignments and tests.

References

[1] J. Hooper, M. Pollanen, and H. Teismann, “Effective Online Office Hours in the Mathematical Sciences”, Journal of Online Learning and Teaching, pp. 187–194, 2, 2006.

[2] L. Padovini, and R. Solmi, “An Investigation on the Dynamics of Direct-Manipulation Editors for Mathematics”, Lecture Notes in Computer Science, pp. 302–316, 3119, 2004.

[3] M. Pollanen, “Interactive Web-Based Mathematics Communication”, Journal of Online Mathematics and its Applications”, online, 6, 2006.

[4] M. Pollanen, “Improving Learner Motivation with Online Assignments”, Journal of Online Learning and Teaching, to appear in June 2007.

[5] S. Smithies, K. Novins, and J. Arvo, “A Handwriting-Based Equation Editor”, Proceedings of Graphics Interface ’99, pp. 84–91, 1999.

[6] E. Tapia, and R. Rojas, “Recognition of On-Line Handwritten Mathematical Formulas in the E-Chalk System”, Proceeding of the Seventh International Conference on Document Analysis and Recognition, pp. 980–984, 2003.

[7] R. Zanibbi, D. Blostein, and J.R. Cordy, “Recognizing Mathematical Expressions Using Tree Transformation”, IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 1455–1467, 24, 2002.